TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových inženýrských studií

Studijní program: B2646 Informační technologie

Studijní obor: Informační technologie

Automatizace tvorby grafu v programu Grapher využívající rozhraní k databázi

Automating of Graphs Creation in Grapher Program Using the Interface to the Database

Bakalářská práce

Autor: Michal Běloch

Vedoucí práce: doc. Ing. Milan Hokr, Ph.D.

V Liberci dne 17. 5. 2013

ZADÁNÍ PRÁCE

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 17. 5. 2013

Podpis

Poděkování

Na tomto místě bych chtěl poděkovat panu doc. Ing. Milanu Hokrovi, Ph.D., vedoucímu mojí bakalářské práce, za odborné vedení, které mi při psaní práce poskytl.
Dále děkuji své rodině a přátelům za všeobecnou podporu během studia.

Anotace

Automatizace tvorby grafu v programu Grapher využívající rozhraní k databázi

Bakalářská práce se zabývá automatickou tvorbou grafu využívající rozhraní k databázi. Automatizace je vytvářena z dat naměřených v tunelu Bedřichov.

První část této práce seznamuje s programem Grapher, jeho možnostmi připojení k databázi, tvorbě grafů a automatizace. Dále seznamuje s obsahem měření v tunelu Bedřichov a vysvětluje pojem databáze.

V praktické části se zabývá vytvořením připojení do databází PostgreSQL a Microsoft Access a vytvořením ODBC spojení s programem Grapher. Dále se zabývá tvorbou a použitím naprogramovaných skriptů, sloužících k automatizaci tvorby grafů.

Klíčová slova: Automatizace tvorby grafu, program Grapher, tunel Bedřichov, databáze PostgreSQL

Annotation

Automating of Graphs Creation in Grapher Program Using the Interface to the Database

This bachelor's thesis deals with automated creation of graphs using interface of database. The automation is created using data collected in the Bedřichov tunnel.

First part of this thesis introduces the Grapher program, its possibilities of connection to the database, creation of graphs and automation. Further it introduces steps of data measurement in the Bedřichov tunnel and explains the database term.

In the practical part the thesis describes connecting into PostgreSQL and Microsoft Access databases and making an ODBC connection to the Grapher program. Further on it describes writing and use of programmed scripts, used in automation of graph making.

Keywords: Automating of graphs creation, Grapher program, Bedřichov tunnel, database PostgreSQL

Obsah

Seznam použitých zkratek

ODBC	Open Database Connectivity
OM	Object Manager
WM	Worksheet Manager
SŘBD	Systém řízení báze dat
SQL	Structured Query Language
API	Application Programming Interface
TUL	Technická univerzita v Liberci
HW	Hardware
GSM	Globální systém pro mobilní komunikaci
TCP/IP	Transmission Control Protocol / Internet Protocol
CSV	Comma-Separated Values
XML	Extensible Markup Language
DSN	Data Source Name

1 Úvod

Dnes se běžně setkáváme s programy, které uchovávají množství dat v databázi i s programy co tvoří jednoduché grafy. Vedle toho se setkáme se specializovanými programy, kde uživatel pracuje se statickými daty. V mé práci se snažím spojit tyto technologie, což sice některé programy umožňují, ale není to rutinně hromadně rozšířené a vyžadují individuální přizpůsobení uživatele či programátora.

Program, ve kterém budu zpracovávat automatizaci, vychází z potřeb měření v tunelu Bedřichov a také navazuje na můj bakalářský projekt, který jsem pojal jako rešerši na téma Grapher, Excel, MatLab a jejich možnosti připojení k databázi [3]. Seznámil jsem s v ní s možnostmi jednotlivých programů v automatizaci a jejich spojením s databází.

Na začátku mojí bakalářské práce jsem se naučil pracovat s programem Grapher od firmy (Golden Software) který se používá pro tvorbu vědeckých grafů. S možnostmi tvorby 2D a 3D grafů a jejich vlastnostmi, možnostmi čar, vyplnění grafu, rozsahy os a možností jejich automatizace, dále se podrobněji seznámil s programovacím jazykem WinWrap Basic, který je v tomto programu využíván pro tvorbu skriptů.

Cílem bakalářské práce je nakonfigurovat ODBC rozhraní, pomocí kterého se připojí a stáhne z databáze naměřená data a vytvořit skripty, které zobrazují grafy z hodnot naměřených v databázi tunelu Bedřichov. Skripty by měly po komunikaci s uživatelem čáry v grafu upravit a poté uložit úpravu grafu do souboru s nastavením, které si může uživatel kdykoliv načíst.

2 Program Grapher

2.1 Úvod do programu Grapher

Grapher je sofistikovaný grafický program, který převádí data do jakéhokoliv typu grafu. Distribuovaný firmou Golden Software. Program je určen pro vědce, vývojáře a manažery obchodů, kteří potřebují zobrazit v grafu své údaje. Grapher vytváří více než 50 různých typů grafů. Od 2D, 3D, polárních a speciálních grafů po vrstevnicové a povrchové mapy. Nejnovější verze programu je Grapher 9. Jeho cena je 349 amerických dolarů [5].

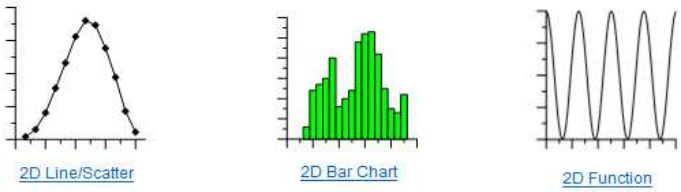
Nástroj pro automatizaci tvorby grafů, které Grapher používá, je takzvaný Scripter, se kterým se seznámíme v kapitole 2.4.

2.2 Typy grafů

Grapher vytváří několik unikátních 2D nebo 3D typů. Příklady některých typů jsou uvedeny v podkapitolách 2.2.1 až 2.2.4.

2.2.1 2D XY grafy

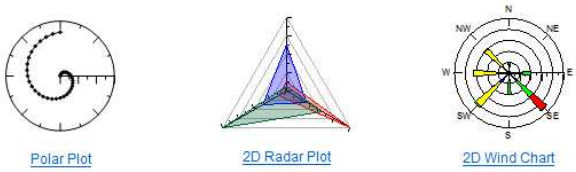
2D XY jsou dvourozměrné plochy, které zobrazují data jako čáry, body nebo bary na 2 osách. Všechny vlastnosti grafu jsou upravovatelné včetně os zobrazující symboly a čáry a u barů šířku pruhů. Tyto grafy budu pro automatizaci využívat nejvíce, z hlediska jejich dobře upravitelných vlastností.



Obr. 1 - 2D grafy

2.2.2 Polární grafy

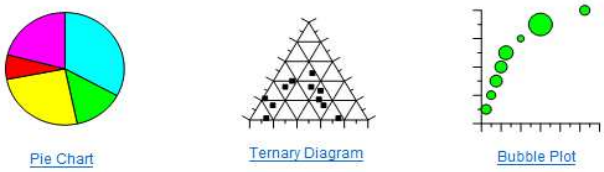
Polární grafy zobrazují data jako čáry, body a bary na polárních osách. Údaje jsou definovány úhly a vzdáleností od středu grafu. Existují různé možnosti pro každý typ grafu, včetně nastavení vlastností os, vlastností čar, vlastností symbolů a barových vlastností.



Obr. 2 - Polární grafy

2.2.3 Speciální grafy

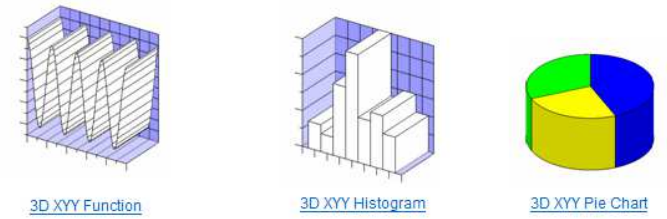
Speciální grafy obsahují mnoho různých typů grafů. Různé vlastnosti, včetně vlastností os, vlastností čar, vlastnosti šířky barů a počet řezů.



Obr. 3 – Speciální grafy

2.2.4 3D XYY grafy

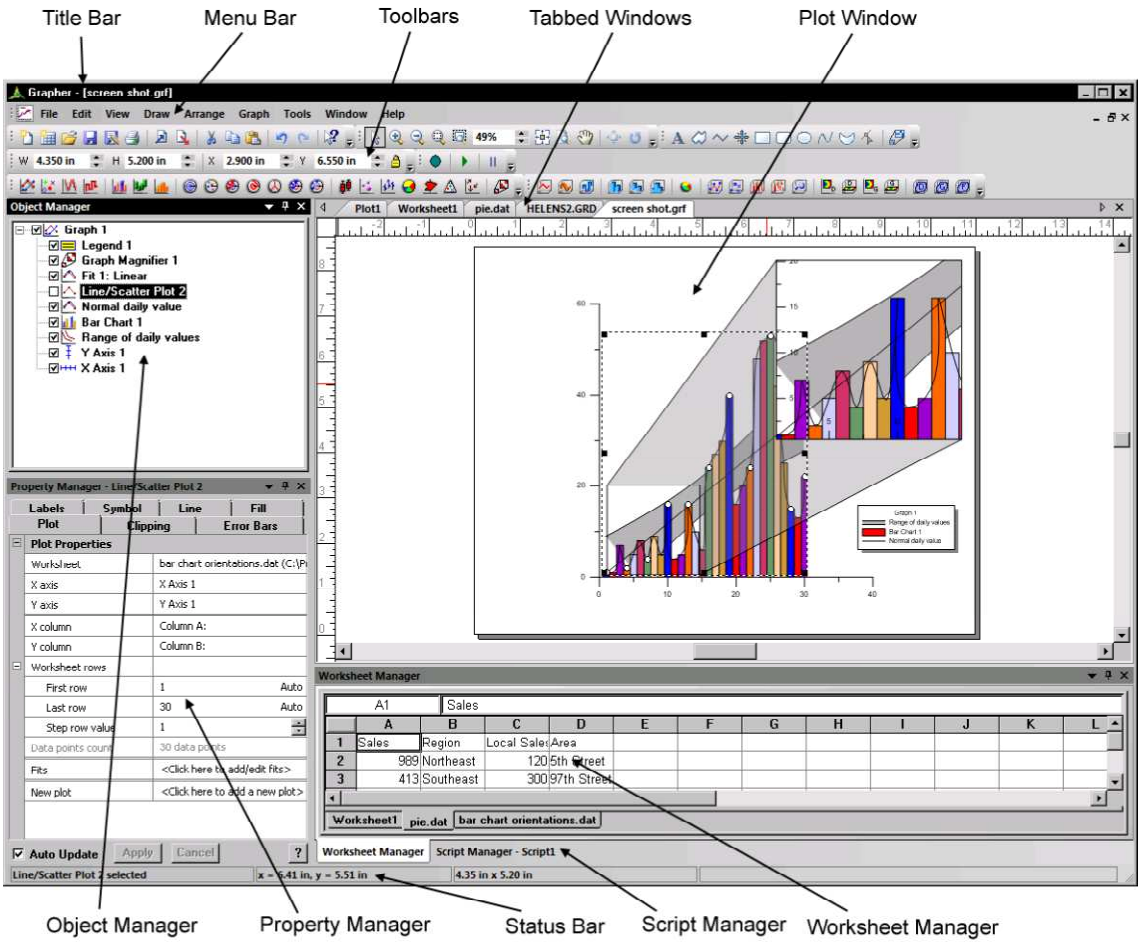
3D XYY grafy zobrazují dvourozměrné údaje ve třech rozměrech. To se provede nastavením třetího rozměru grafu do šířky. Všechny vlastnosti, včetně vlastností čar, vlastností výplně, vlastností os a šířkou grafu jsou nastavitelné.



Obr. 4 - 3D XYY grafy

2.3 Uživatelské rozhraní programu Grapher

Uživatelské rozhraní programu Grapher je velice příjemné a skládá z Title Bar, Menu Bar, Toolbars, Tabbed Windows, Object Manager a Status Bar. Grapher obsahuje 4 typy okenních dokumentů: Plot Window, Worksheet Manager, Excel worksheet Windows a Grid Windows. Grafy a mapy se zobrazují a mohou být upraveny v Plot Window. Worksheet Windows zobrazuje, upravuje, převádí a ukládá data tabulkového formátu. Excel worksheet Windows umožňuje otevřít původní Excel okno v programu Grapher. Jednotlivé komponenty popisuji v Tabulka 1.



Obr. 5 - Uživatelské rozhraní programu Grapher

Tabulka 1 - Funkce komponent programu Grapher

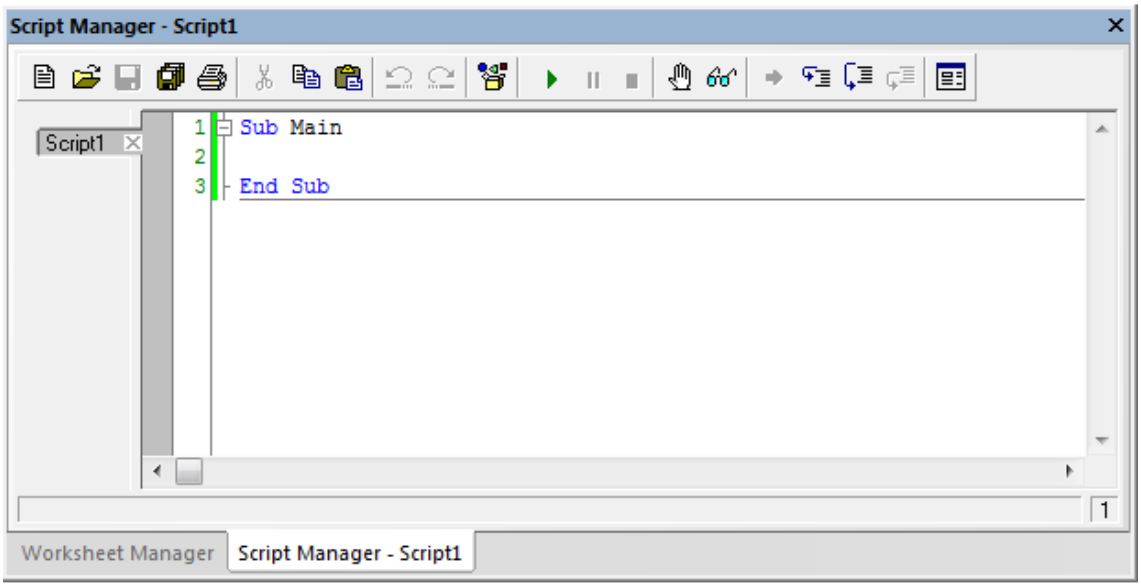
Jméno komponenty	**Funkce komponenty**
Title Bar	Záhlaví obsahuje název programu plus uložený název, pokud existuje. Hvězdička za názvem souboru označuje, že soubor byl upraven před posledním uložením.
Menu bar	Menu lišta obsahuje používané příkazy ve spuštěném programu Grapher
Toolbars	Panel nástrojů obsahuje tlačítka nástrojů programu Grapher, které jsou zkratky menu příkazů.
Tabbed Window	Vícenásobný plot Windows, worksheet Windows, Excel Windows a Grid Windows se zobrazují jako záložky. Kliknutím na záložku se zobrazí obsah okna.
Object Manager	Object Manager obsahuje hierarchický seznam všech objektů obsažených v Plot Window. Tyto objekty mohou být vybrány, přidány, uspořádány upraveny a přejmenovány v Object Manager. OM je původně ukotven na levé straně nad Property Manager. Změny provedené v OM se okamžitě projeví v Plot Window.
Property Manager	Property Manager umožňuje upravovat některou z vlastností vybraného objektu. Více objektů může být naráz upraveno vybráním všech objektů a změnou sdílených vlastností. Změny provedené v Property Manager se ihned projeví v Plot Window.
Worksheet Manager	Worksheet Manager obsahuje přehled o všech datech nahraných do programu Grapher. Úpravy provedené ve WM se automaticky promítnou v grafu.
Script Manager	Správce skriptů kontroluje nahrané a běžící skripty v programu Grapher.
Status Bar	Stavový řádek zobrazuje informace o činnostech v programu Grapher. Stavový řádek je rozdělen do tří částí, které obsahují informace o vybraných příkazech, objektech nebo pozicích.

2.4 Scripter

Scripter, jako součást programu Grapher, je vhodný pro vytváření, editaci a provádění skriptů, které automatizují operace programu Grapher. S využitím skriptů mohou být jednoduché každodenní, ale i komplexní úkoly prováděny opakovaně bez přímé interakce [1].

Script recorder zaznamenává všechny příkazy prováděné uživateli v programu Grapher. Když je skript spuštěn, Grapher provede zaznamenané kroky. Což je vhodné pro uživatele, kteří často provádí stejné úkoly, ale nejsou zběhlí v automatizaci, pro pokročilé uživatele, kteří mají problémy se syntaxí nebo ji pouze nechtějí celou zadávat ručně.

V mé bakalářské práci budu ve Scripteru vytvářet a pracovat se skripty pro připojení k databázi a budu ho využívat pro automatizaci tvorby grafů.



Obr. 6 – Grafické rozhraní Scripteru

Scripter si v programu Grapher odkryjeme v liště menu v možnosti View, Managers a zaškrtneme Script Manager.

Vrchní část je zobrazený Toolbar, kde jsou možnosti, které vytvoří nový skript, otevřou uložený skript, tisknou, spustí skript. V levé části jsou aktuálně otevřené skripty. Zbylá část, je pracovní okno, ve kterém se píší skripty. Scripter je napsán ve WinWrap Basic Language a k programování skriptů používá Visual Basic for Applications(TM) kompatibility.

V programu Grapher, Scripter používá knihovnu obsahující data o všech typech grafů, worksheetu, možnostech ukládání, načítání grafů a vlastnostech jednotlivých os grafů. Knihovna není přístupná odjinud, než z programu Grapher.

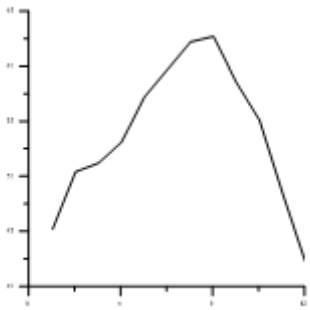
2.4.1 Ukázka a popis skriptu pro automatickou tvorbu grafu:

Ukázka jednoduchého vytvoření grafu pomocí Skripteru za použití nejnutnějších příkazů.

```
Sub Main
'Vytvoří Grapher jako objekt
Dim Grapher As Object
'Spustí Grapher
Set Grapher = CreateObject("Grapher.Application")
Grapher.Visible = True
'Vytvoří nové okno dokumentu
Set Plot1 = Grapher.Documents.Add(grfPlotDoc)
'Vytvoří osy grafu z datového souboru
Plot1.Shapes.AddLinePlotGraph("C:\Program      Files\Golden
Software\Grapher 9\Samples\Tutorial.dat",1,2)
'Definuje objekt ukazatele
Set Graph1 = Plot1.Shapes.Item(1)
'Definenuje objekt ukazatele
Set LineScatterPlot1 = Graph1.Plots.Item(1)
'Vybere objekt
LineScatterPlot1.Select
End Sub
```

Výsledný graf:

Na Obr. 7 je zobrazen jednoduchý vytvořený 2D Plot, kde je pouze Xová a Yová osa a v grafu je zobrazeno pouze 10 bodů.



Obr. 7 - Ukázkově vytvořený graf v programu Grapher

3 Teorie databází

Databáze je určitá uspořádaná množina informací uložená na nějakém paměťovém médiu [4]. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se v české odborné literatuře nazývá systém řízení báze dat (SŘBD).

3.1 Historie

Předchůdce databází byly papírové kartotéky. Správa takový kartoték byla v mnohém podobná správě dnešních databází.

Velký impulz pro rozvoj databází byl vývoj počítačů v 50 letech 20 století. Ukázalo se, že původně univerzální používání strojového kódu procesorů je (nejen) pro databázové úlohy neefektivní, a proto se objevil požadavek na vyšší jazyk pro zpracování dat.

To mělo za následek vytvoření vyšších programovacích jazyků. V roce 1962 jazyk COBOL a v roce 1965 jazyk PL/1.

V roce 1970 začínají zveřejněním článku E. F. Codda první **relační databáze**, které pohlížejí na data jako na tabulky. Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL.

V 90. letech 20. století se začínaly objevovat první **objektově** orientované databáze, jejichž filozofie byla přebírána z objektově orientovaných jazyků.

Pojem „databáze“ je často zjednodušován na to, co je ve skutečnosti databázový systém (databázový stroj) nebo též systém řízení báze dat. Ten neobsahuje pouze **tabulky** – ty jsou jedny z mnoha tzv. databázových objektů (někdy též databázových entit). Pokročilejší databázové systémy obsahují například:

- **pohledy** neboli **views** – SQL příkazy, pojmenované a uložené v databázovém systému. Lze z nich vybírat (aplikovat na ně příkaz SELECT) jako na ostatní tabulky.
- **klíče** neboli **indexy** pro každou tabulku. Klíče jsou definovány nad jednotlivými sloupci tabulek (jeden klíč jich může zahrnovat i více) a jejich funkce je vést si v tabulkách rychlé LUT (*look-up tables* – „pořadníky“) na sloupce, nad nimiž byly

definovány, vyloučit duplicitu v záznamech nebo zajišťovat fulltextové vyhledávání.

- **spouště** neboli **triggery** – mechanismus mezi řádky dvou tabulek, který se v databázovém systému dá definovat jako jeden z několika úkonů, který se vyvolá po změně nebo smazání rodičovské tabulky.
- **procesy** – databázové stroje umí podat přehled o procesech, které jejich služeb aktuálně využívají.

3.2 Relační databáze

Základním konstruktorem relačních databází jsou relace (databázové tabulky), což jsou dvourozměrné struktury tvořené záhlavím a tělem. Jejich sloupce se nazývají atributy, řádky tabulky jsou pak záznamy. Atributy mají určen svůj konkrétní datový typ a doménu, což je množina přípustných hodnot daného atributu. Řádek je řezem přes sloupce tabulky a slouží k vlastnímu uložení dat [9].

Terminologie:

Kandidátní klíč - Kandidátní klíč je atribut nebo skupina atributů, které jednoznačně identifikují záznam v relační tabulce. Kandidátní klíč se může stát primárním klíčem; ty které se primárním klíčem nestanou jsou označovány jako alternativní klíče.

Primární klíč - je jednoznačný identifikátor záznamu, řádku tabulky. Primárním klíčem může být jediný sloupec či kombinace více sloupců tak, aby byla zaručena jeho jednoznačnost. Pole klíče musí obsahovat hodnotu, tzn. nesmí se zde vyskytovat nedefinovaná prázdná hodnota NULL.

Cizí klíč - Dalším důležitým pojmem jsou nevlastní/cizí klíče. Slouží pro vyjádření vztahů, relací, mezi databázovými tabulkami. Jedná se o pole či skupinu polí, která nám umožní identifikovat, které záznamy z různých tabulek spolu navzájem souvisí.

Vztahy mezi tabulkami:

Vztahy (relationships) slouží ke svázání dat, která spolu souvisejí a jsou umístěny v různých databázových tabulkách. Každý vztah je charakterizován třemi základními vlastnostmi:

- stupněm
- kardinalitou
- volitelností účasti

Stupeň vztahu

1. Unární vztah - relace je spojena sama se sebou. Typickým příkladem je vztah Zaměstnanec - Nadřízený, kdy nadřízený je také jedním ze zaměstnanců a může mít také nadřízeného. Vztah se realizuje vložením primárního klíče relace Zaměstnanec ve formě cizího klíče opět do relace Zaměstnanec.
2. Binární vztah - klasický vztah mezi dvěma relacemi.
3. Ternární vztah - jedná se o vztah mezi třemi relacemi najednou.
4. Enární vztah - jedná se vztah mezi n-relacemi zároveň.

Ternární a enární vztahy se nesnadno modelují a v praxi se objevují velice zřídka.

Kardinalita vztahu

1. mezi daty v tabulkách není žádná spojitost, proto nedefinujeme žádný vztah.
2. 1:1 používáme, pokud záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak. Takovýto vztah je používán pouze ojediněle, protože většinou není pádný důvod, proč takovéto záznamy neumístit do jedné databázové tabulky. Jedno z mála využití je zpřehlednění rozsáhlých tabulek. Jako ilustraci je možné použít vztah řidič - automobil. V jednu chvíli (diskrétní časový okamžik) řídí jedno auto právě jeden řidič a zároveň jedno auto je řízeno právě jedním řidičem.
3. 1:N přiřazuje jednomu záznamu více záznamů z jiné tabulky. Jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě. Jako reálný příklad může posloužit vztah autobus - cestující. V jednu chvíli cestující jede právě jedním autobusem a v jednom autobuse může zároveň cestovat více cestujících.

4. M:N je méně častým. Umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. V databázové praxi bývá tento vztah z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné tabulky složené z kombinace obou použitých klíčů (třetí resp. tzv. vazební tabulka). Příkladem z reálného života by mohl být vztah výrobek - vlastnost. Výrobek může mít více vlastností a jednu vlastnost může mít více výrobků. V reálném životě nicméně existuje velké množství vztahů M : N, mimo jiné také proto, že často existuje praktická potřeba zachovávat i údaje o historii těchto vztahů z časového hlediska (jeden řidič v delším časovém období řídí více rozličných aut a jedno auto v delším časovém období může mít více různých řidičů).

Volitelnost účasti ve vztahu

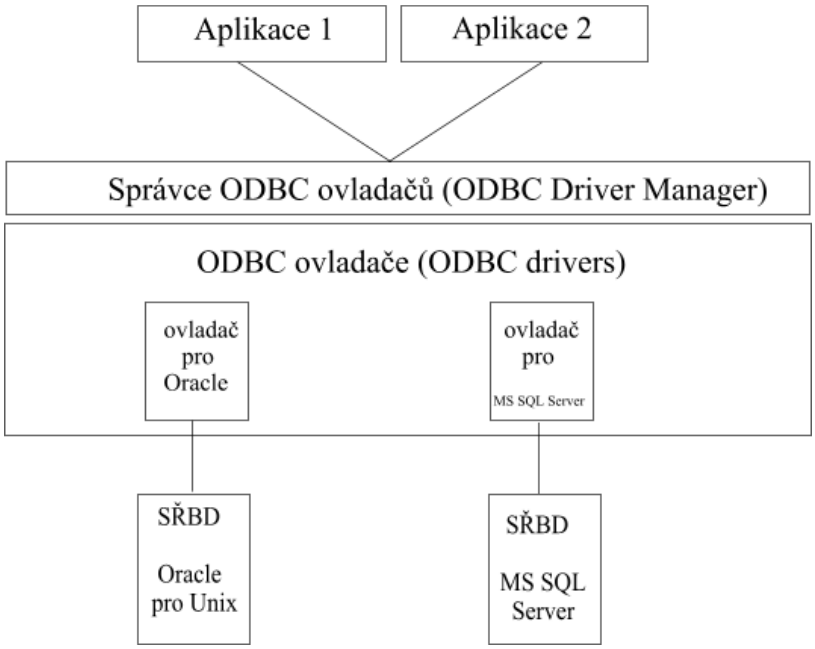
Volitelnost účasti ve vztahu vyjadřuje, zda je účast relace ve vztahu povinná nebo volitelná.

3.3 Open Database Connectivity

ODBC (Open Database Connectivity) – je programovací rozhraní, které aplikacím umožňuje přístup k datům v systémech řízení bází dat, které jako standard pro přístup k datům používají jazyk SQL (Structured Query Language) [8].

Použitím API ODBC mohou aplikace přistupovat k datům, nezávisle na tom, jakým systémem pro řízení báze dat (dále jen SŘBD resp. DBMS) jsou tato data spravována a to i přesto, že každý SŘBD používá pro uložení dat jiný formát a jiné programové rozhraní. Zavedení ODBC, jakožto standardu, pro přístup k datům se ukázalo být velmi prozíravým činem, který přinesl mnoho výhod nejen programátorům (resp. koncovým uživatelům), ale i samotným výrobcům systémů pro řízení báze dat. Přestože za tvorbou ODBC stojí společnost Microsoft, tak použití ODBC není nijak vázáno pouze na platformu Win32.

Architektura ODBC na Obr. 8



Obr. 8 – Schéma ODBC

Model struktury ODBC se dá znázornit pomocí čtyř vrstev.

- **V první nejvrchnější vrstvě** se nachází samotná aplikace. Ta v případě, že potřebuje data, provede volání ODBC funkcí (ve formě SQL dotazu).
- **Druhou vrstvou** je tzv. „Správce ODBC ovladačů“ (ODBC Driver Manager). Úkolem správce ovladačů je zajistit propojení mezi aplikací a příslušným ODBC ovladačem (ODBC ovladače tvoří třetí vrstvu modelu, podrobněji viz dále). Jakmile aplikace potřebuje data, správce ovladačů vyhledá a nahraje příslušný ovladač. (ve formě DLL knihovny). Správce ovladačů také zjistí jaké konkrétní funkce jsou podporovány jednotlivými ovladači a uschová si jejich adresy v paměti do tabulky. V případě, že aplikace volá konkrétní funkci, správce souborů zjistí, ke kterému ovladači funkce patří a zavolá ji. Tímto způsobem může být prováděn souběžný přístup k více ovladačům, což se hodí v případě programování aplikací přistupujících souběžně k několika zdrojům dat.
- **Třetí vrstvou** zde již zmíněnou vrstvou jsou ODBC ovladače. Ty provedou zpracování volané ODBC funkce, přeložení požadavku do SQL pro příslušný SŘBD (DBMS) a jeho následné poslání.
- **Poslední vrstvou** je SŘBD, který provede zpracování operace požadované ODBC ovladačem a výsledky této operací mu vrátí.

Hlavní a zásadní výhodou plynoucí z použití ODBC je zjednodušení přístupu do databáze. ODBC zavádí abstrakci nazvanou „data source“ (datový zdroj). Pod pojmem „data source“ si lze představit nějaké smysluplné symbolické jméno pro zdroj dat např. platba, inventář apod. ODBC pak toto jméno přiřazuje přes konkrétní ovladač, síťový software, jméno serveru nebo adresu do SŘBD. Přitom je úplně jedno, zda se zdroj dat nachází v lokální databázi nebo někde na síti. Programátor je tímto zbaven povinnosti zabývat se jakýmikoliv síťovými záležitostmi.

4 Tunel Bedřichov

4.1 Přírodní podmínky

Lokalita Bedřichovský tunel se nachází v Jizerských horách v severních Čechách, mezi obcemi Bedřichov a Janov a nachází se v něm vodovodní potrubí.

Tunel byl ražen v letech 1979 – 1981, měří 2 600 m, průměr je 3,3 m a vodovodní potrubí má průměr 80 cm. Výzkum zde začal v letech 2002 – 2003 zakázkami od SÚRAO pro ČGS.

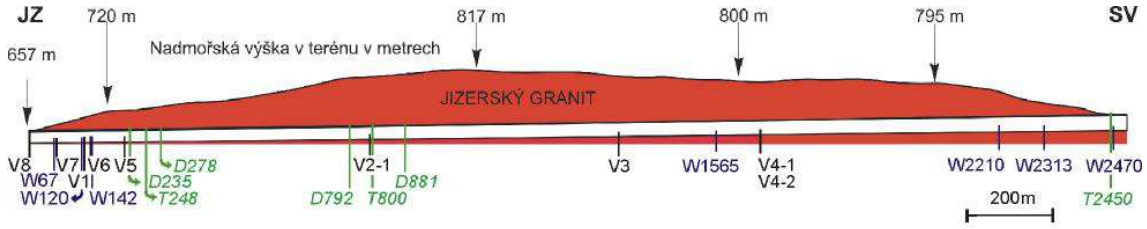
- 2002 – 2003 geologická a strukturní charakteristika granitoidů z tunelu Bedřichov v Jizerských horách
- 2004 – 2005 geologická a strukturní charakteristika granitoidů z vodárenských tunelů v Jizerských horách.
- 2006 – 2008 studium dynamiky puklinové sítě granitoidů ve vodárenském tunelu Bedřichov v Jizerských horách.
- Od roku 2009 převzala práce v tunelu Bedřichov TUL.

4.2 Obsah měření

V tunelu je od roku 2009 budován systém pro automatický sběr dat [6]. Data zaznamenané senzory uvnitř tunelu jsou přenášena na vzdálený server, kde se ukládají do databáze. V tunelu se provádějí např. průtoková a teplotní měření, seismická měření [7].

4.2.1 Monitorování

Do monitorovacího programu patří sada pravidelně monitorovaných pramenů. Místa monitorovaných podzemních pramenů jsou označena na stěně tunelu A písmenem V a pořadovým číslem. Místa nově monitorovaných pramenů byla označena písmenem W a pozicí pramenu (v metrech od portálu tunelu). Všechny pravidelně monitorované prameny jsou uvedeny níže a jejich pozice je zobrazena na Obr. 9



Obr. 9 - Schéma tunelu Bedřichov.

Přehled měřených veličin, způsob frekvence měření a odběru vzorků na jednotlivých pramenech přidávám v Příloze A – Seznam použitých měřících zařízení v Tunelu Bedřichov.

4.2.2 Přenos dat z tunelu Bedřichov

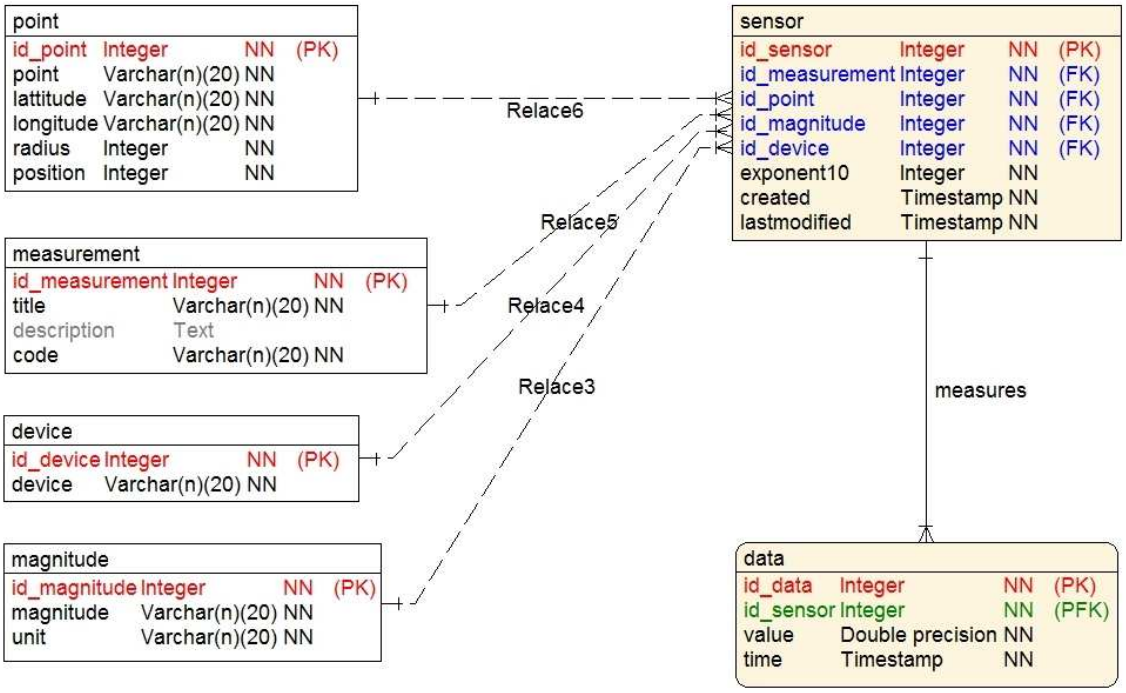
Přenos spočívá v odesílání dat naměřených v určité lokalitě na databázový server, kde proběhne jejich uložení. Jednotlivé měřící přístroje v lokalitě jsou propojeny sběrnicí RS485 k zařízení s HW modulem TC65i. Zařízení sbírá data a pak odesílá na server [10].

Systém se skládá z:

- Senzorů umístěných v tunelu
- Řídící a komunikační systému s nízkonapěťovým microkontrolerem a GSM modulem u vstupu do tunelu komunikujícím s umístěnými senzory a vysílající shromážděná data přes bezdrátovou síť
- Aplikace na serveru (zvaná Remote Data Receiver) zodpovědná za:
    - Přijímaní TCP/IP paketů
    - Získávání naměřených dat uložených ve EPSNet paketech (originální pakety se užívají pro komunikaci v tunelu)
    - Dekódovat data z EPSNet paketů a ukládat je do struktury relační databáze
- Databázový server věnující se ukládání naměřených dat do struktury navržené relační databáze

4.3 Databáze k tunelu Bedřichov

Data monitorované v Tunelu Bedřichov, se ukládají na databázový server. Tento databázový server je umístěn na portále [2]. Díky tomu je možné získat přehled všech senzorů a naměřených dat. Pro účely mojí bakalářské práce poslouží databáze k Tunelu Bedřichov jako předloha pro vytvoření vlastní databáze. Schéma databáze je na Obr. 10



Obr. 10 – Schéma navržené databáze k tunelu Bedřichov

Popis atributů v databázových tabulkách:

Tabulka Data: Obsahuje naměřenou hodnotu, senzor, na kterém byla naměřena a čas.

Tabulka Senzor: Obsahuje Název měření, Bod měření, přístroj a veličinu

Tabulka Bod: Obsahuje název bodu, latitude, longitude, radius a pozici.

Tabulka Měření: Obsahuje Název a popis měření.

Tabulka Přístroje: Obsahuje název přístroje a číslo přístroje.

Tabulka Veličina: Obsahuje Název veličiny a její jednotku.

4.3.1 Vztahy mezi tabulkami v databázi

V tabulce data je obsažen id_sensor, což je cizí klíč. Touto hodnotu se přiřazuje senzor, kterým byla data naměřena.

Struktura databáze je vysvětlena v Tabulka 2 – Tabulka 6 :

Tabulka 2 - Tabulka Bod bude obsahovat všechny body, kde jsou nainstalované zařízení a na nich konkrétní senzory.

Id_point	point	latitude	longitude	radius	position
1	Bod měření 1	10	80	30	1520
2	Bod 2	30	50	40	1360

Tabulka 3 - Tabulka Zařízení obsahuje seznam použitých senzorů v dole. Pojem Device zde označuje senzory, které provádí měření.

Id_device	Device_cs
1	Senzor teploty
2	Senzor průtoku
3	Hladinoměr

Tabulka 4 - Tabulka Veličina obsahuje veličiny a jejich jednotky, které se používají při měření v dole.

Id_magnitude	Magnitude_cs	Unit_cs
1	Metry krychlové	m3
2	Metry čtvereční	m2
3	Teplota	°C

Tabulka 5 - Tabulka Měření obsahuje názvy a popis měření, která se doposud prováděla v dole.

Id_measurement	Title_cs	description	Code_cs
5	Import dat z obecných zdrojů	Data, která přichází od zařízení nepodporující komunikační protokol tunelu	ID
10	Water log		WT1

Tabulka 6 - Tabulka Senzor.

Id_sensor	Id_measurement	Id_point	Id_magnitude	Id_device	exponent	created	lastmodified
1	5	1	2	1	-	-	-
2	10	2	1	2	-	-	-

Tabulka Sensor obsahuje 4 cizí klíče. Id_magnitude, přiřazuje veličinu z tabulky Veličina k naměřené hodnotě na senzoru, id_device přiřazuje zařízení z tabulky Zařízení, id_measurement přiřadí měření z tabulky Měření a id_point přiřazuje konkrétní bod z tabulky Bod.

Tabulka data může obsahovat vícekrát hodnotu od stejného senzoru.

V tabulce Senzor bude odkazovat na konkrétní bod a zařízení, na kterém byla data naměřena, bude obsahovat název měření a v jaké veličině, byla data přijata.

Tabulka 7 - Výsledná tabulka po spojení

Id_sensor	Title_cs	point	Magnitude_cs	Device_cs	exponent	created	lastmodified
1	Import dat z obecných zdrojů	Bod měření 1	m2	Senzor teploty	-	-	-
2	Water log	Bod 2	m3	Senzor průtoku	-	-	-

5 Vlastní část práce

Pro mou práci jsem měl k dispozici kopii databáze Tunelu Bedřichov v PostgreSQL. Bohužel na začátku práce se mi nedařilo propojit program Grapher s databází PostgreSQL, proto jsem se rozhodl vytvořit totožnou databázi v Microsoft Access, databázovém programu, kde mi spojení s programem Grapher fungovalo. V další části jsem se rozhodl otestovat připojení k MySQL databázi, jenže jsem měl stejný problém jako u PostgreSQL. Chybu v automatickém připojení k databázi jsem nahradil připojovacím řetězcem ve skriptu a díky tomu jsem se připojil do databází MySQL i PostgreSQL. Pokračoval jsem tedy ve vypracování v PostgreSQL.

V části práce budu demonstrovat vytvoření ODBC připojení do použitých typů databází. Stažení dat z databází do programu Grapher pomocí vytvořených skriptů a ODBC zdrojů dat. PostgreSQL jsem testoval pro stažení dat ze Serveru mimo lokální síť. Použil jsem verzi Windows Server 2012 Standard Evaluation (trial verzi na 180 dní).

V programovací části jde o tvorbu skriptů, pro automatizaci připojování k databázím a vykreslování grafů, možnost ukládat a načítat nastavení pro vytvořené grafy.

Během tvorby bakalářské práce jsem se setkal s nekompatibilitou softwaru. Práci jsem vypracovával na počítači s nainstalovaným 64bitovým systémem a s nainstalovanou 64bitovou verzí programu Grapher. Bohužel ODBC rozhraní mi na 64bitovém systému nedovolilo spravovat (*.mdb) zdroj dat. Toto omezení mi nedovolilo načítat data z databáze. Proto jsem musel použít 32bitový operační systém, 32bitovou verzi programu Grapher. V takovéto kombinaci jsem už žádné problémy, s načítáním dat z databáze pomocí ODBC neměl. Použil jsem proto Virtual Box od firmy Oracle, do kterého jsem nainstaloval 32bitovou verzi Windows 7.

5.1 Tvorba a přístup k databázím

5.1.1 Vytvoření databáze v Microsoft Access 2007

Databáze je naplněna daty z tunelu Bedřichov. Použita byla studentská verze Microsoft Office stažen z webu MSDN.

Použité datové typy v databázi:

- *Automatické číslo* – Je to privátní klíč (jedinečná hodnota) hodnota začínající 1. až po počet záznamů v tabulce. Hodnota se v záznamech nemůže opakovat.
- *Číslo* – V Microsoft Access označeno jako dvojitá přesnost na 15 desetinných míst (float)
- *Text* – počet znaků, které se do pole dají vepsat, je 255.
- *Datum a čas* – obecné datum ve formátu 1.12.2012 19:45:22

Vkládání dat do tabulky

Pro naplnění záznamů v tabulkách používám dva způsoby:

1. **Ruční**

   Vytvářím nové záznamy, ve vizuálním prostředí Accessu, vkládáním hodnot do jednotlivých řádků databázové tabulky. Při vložení záznamu do jakéhokoliv řádku databázové tabulky se automaticky vytvoří Automatické číslo ve sloupci ID_Data.

   Skripty testuji na záznamech takto vytvořených. Šlo o vyladění napsaných skriptů v programu Grapher, protože vím, jaké záznamy jsou v databázi a jaké mohu čekat výsledné grafy. Mohl jsem pracovat na skriptech, aniž bych používal data z databáze PostgreSQL.

   Všimnul jsem si menšího problému formátování při práci v Accessu, Mnou používaný byl nastavený pro oblast používání v ČR, kde desetinná místa oddělujeme čárkou. V Americe (program Grapher) desetinné místo odděluje tečkou.

2. **Import ze souboru XML**

   Souborem XML, lze zaplnit tabulku „Data“, ze které se v programu Grapher načítají hodnoty pro vytvoření grafu. Pro načtení souboru XML se v programu Access musí přepnout do záložky Externí data a poté vybrat tlačítko „Soubor XML“. Dojde k zobrazení formuláře, ve kterém vyberu cestu k souboru. Po načtení souboru se musí nastavit, jak a jaké hodnoty chceme do tabulky Data importovat. Můžeme vytvořit novou tabulku „Data1“ nebo připojit k existující tabulce.

Z důvodu formátu číselných hodnot vkládaných do databázové tabulky, popsané v kapitole o ručním vkládání dat, je vhodné data uložit do nové tabulky „Data1“. Všechny sloupce se nastaví na datový typ Text. Nejjednodušší způsob jak přenést potřebné data do tabulky Data je, že ve sloupci Hodnota (ve vytvořené tabulce z importu) nahradíme všechny tečky za čárky a záznamy potom přeneseme do tabulky „Data“.

Při spuštění skriptu pro porovnání čar z více senzorů, se v první části skript připojí do tabulky „Data1“ a získá informace o jednotlivých senzorech.

Hodnoty pro vytvoření čar pak skript používá z tabulky „Data“.

Uložení vytvořené databáze

Hotovou databázi ukládám jako Tunel Bedřichov.mdb ve formátu „databáze aplikace Access 2002 – 2003“. Vybral jsem si tento formát kvůli konektivitě s ODBC.

Program Grapher při načítání uložené databáze rozeznává datové typy přiřazené jednotlivým sloupcům. Rozezná datový typ Datum a čas a typ Číslo. Jinak nastavené datové typy ve sloupcích vedou ke špatnému zobrazení hodnot a dat na osách v grafu.

5.1.2 Databáze v PostgreSQL

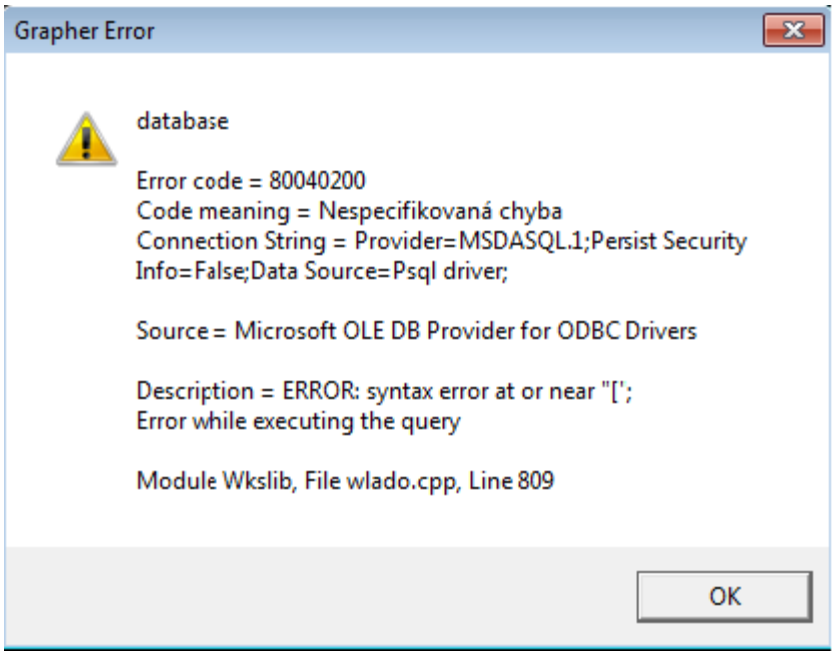
PostgreSQL je relační databázový systém s otevřeným zdrojovým kódem. Běží na všech rozšířených operačních systémech včetně Linuxu, Unixu a Windows. Používal jsem verzi PostgreSQL 9.1, staženou zdarma ze stránek PostgreSQL.

Verzi 9.1 jsem nainstaloval lokálně ve Virtual Boxu a na server Windows 2012. Pro komunikaci na localhostu jsem přidělil port 5433 a na serveru port 5432. V obou případech jsem používal vytvořeného uživatele postgres. Od Ing. Špánka jsem dostal zálohu databáze tunelu Bedřichov, kterou jsem si po úpravách oprávnění některých tabulek importoval do databáze. Import jsem musel provést v SQL Shell příkazem *\i C:/import.sql*

Stejné to bylo s instalací a importem na serveru. Tam jsem ovšem musel ještě upravit soubor pg_hba.conf, aby obsahoval IP adresu, ze které se připojuji. Kdyby tomu tak nebylo, spojení se serverem proběhne úspěšně, ale PostgreSQL odmítne spojení s uživatelem, který se vzdáleně připojuje.

Chyba v programu Grapher při automatickém načítání databáze

V programu Grapher je možnost automaticky načíst databázi, bohužel pro načítání dat z databází PostgreSQL a MySQL je automatický příkaz mapován špatně a při pokusu načtení Grapher oznámí chybu zobrazenou na Obr. 11



Obr. 11 – Chyba při pokusu připojení do databází PostreSQL a MySQL pomocí automatického připojení.

Grapher se automaticky snaží vygenerovat příkaz do databáze, který vypadá přibližně takto: `SELECT * FROM [data]`. Chyba nám napovídá, že se jedná o syntax error. Přičemž by dotaz neměl obsahovat hranaté závorky. Připojení do databáze lze úspěšně docílit použitím vlastního připojovacího řetězce z vytvořeného zdroje dat ODBC.

Připojovací řetězec

Pro připojení k databázi je nutné získat připojovací řetězec z vytvořeného zdroje dat ODBC. Obsahuje informace o názvu DSN, databázi, uživateli i heslu. Jeho pomocí nahrazuji chybně naformátovaný příkaz v programu Grapher takto:

```
DSN=PostgreANSI;DATABASE=Bedrichov;SERVER=localhost;PORT=54
33;UID=postgres;PWD=michal1234;SSLmode=disable;ReadOnly=0;P
rotocol=7.4;FakeOidIndex=0;ShowOidColumn=0;RowVersioning=0;
ShowSystemTables=0;ConnSettings=;Fetch=100;Socket=4096;Unkn
ownSizes=0;MaxVarcharSize=255;MaxLongVarcharSize=8190;Debug
=0;CommLog=0;Optimizer=0;Ksqo=1;UseDeclareFetch=0;TextAsLon
```

```
gVarchar=1;UnknownsAsLongVarchar=0;BoolsAsChar=1;Parse=0;Ca
ncelAsFreeStmt=0;ExtraSysTablePrefixes=dd_;;LFConversion=1;
UpdatableCursors=1;DisallowPremature=0;TrueIsMinus1=0;BI=0;
ByteaAsLongVarBinary=0;UseServerSidePrepare=0;LowerCaseIden
tifier=0;XaOpt=1"
```

5.1.3 Databáze v MySQL

MySQL je relační databáze typu DBMS (database managment systém) a je šířen jako Open Source. MySQL databáze sloužila jako test pro připojování a stahování dat ze serveru, ještě před zprovozněním cílové databáze v PostgreSQL. Díky tomu, že jsem pracoval i na tomto náhradním řešení, jsem odhalil příčinu chyby v připojení do databáze PostgreSQL Programu Grapher.

MySQL jsem si nainstaloval v balíčku Wamp 2.2, který obsahuje Apache server 2.2.21, PHP 5.3.8, PhpMyAdmin 3.1.3.1 a MySQL 5.5.16. Databázi tunelu Bedřichov vytvořenou v PostgreSQL nelze jednoduše nahrát do MySQL, byly by nutné úpravy v importovaném souboru a nebyl to hlavní bod práce. Pro testování připojení do velké databáze mi posloužil import databáze od Ing. Kretschmera o velikosti 1GB (záloha databáze tunelu Bedřihov Ing. Špánka má okolo 200MB).

Pro import takto velké databáze do programu wamp bylo nutné udělat malé úpravy v nastavení pro import souborů, které se běžně pohybují okolo 12MB a doba importu trvá 300 sekund. Velikost vstupního souboru jsem změnil na 2000MB a dobu importu na 0 (neomezeně). Na serveru jsem musel ještě v SQL shellu vytvořit nového uživatele a udělit mu práva pro přístup do MySQL zvenčí.

Připojovací řetězec

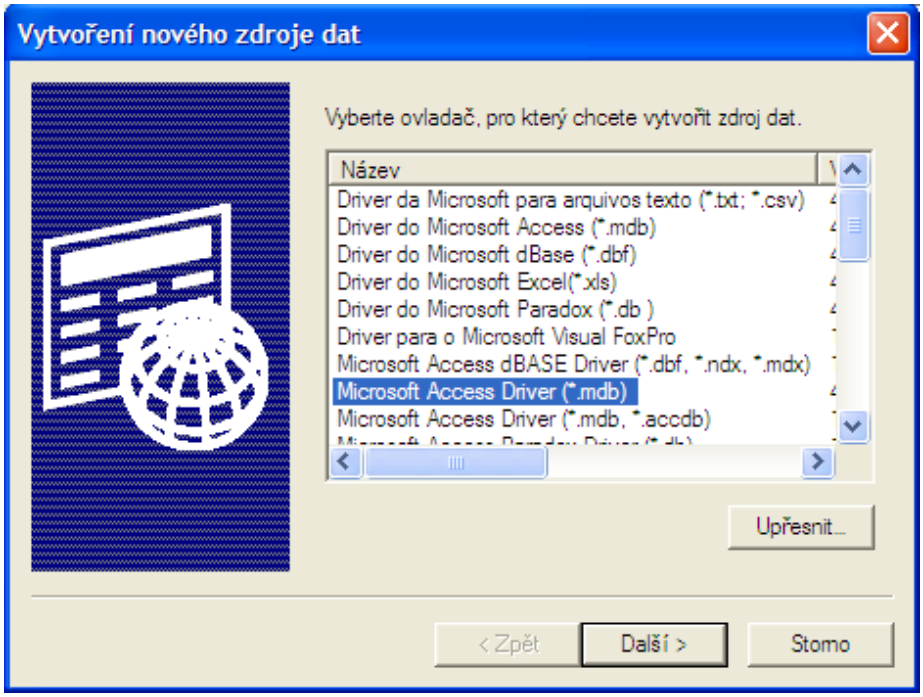
Je nutné ho získat ze zdroje dat pro připojení ODBC.

```
DSN=VZD_david_MySQL;SERVER=78.80.123.80;UID=michal;PWD=Mich
al1234;DATABASE=maredata;PORT=5432
```

5.1.4 Vytvoření zdroje dat pro použité typy databází

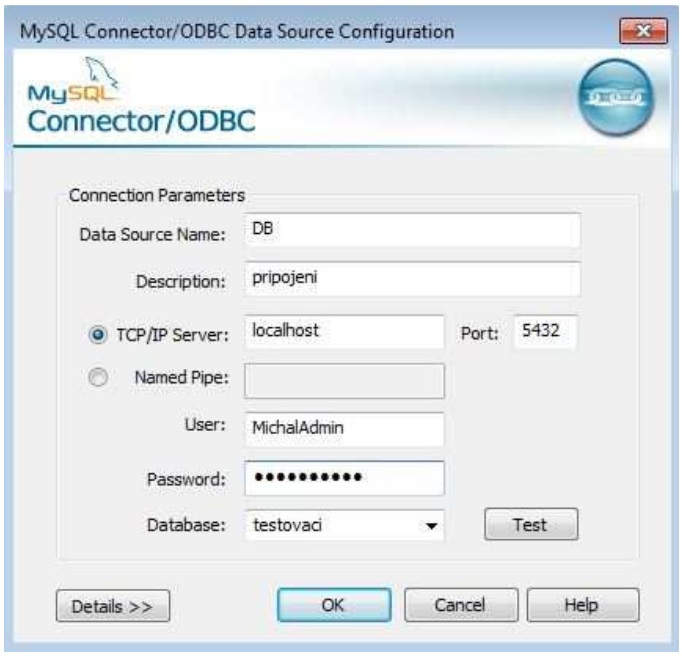
Propojení programu Grapher s databází Microsoft Access realizuji vytvořeným zdrojem dat, který odkazuje na ovladač ODBC. Ve správci zdrojů dat ODBC, v záložce souborové DSN (Data Source Name) je možnost přidat nový zdroj dat Obr. 12. Pro vytvoření spojení jsem použil Microsoft Access Driver (*.mdb) a uložil ho, pod názvem FILE.dsn do složky Data Sources (Zdroje dat).

Obdobně se vytváří zdroj dat pro PostgreSQL a MySQL. U obou se vyplňuje Název, Databáze, Server, Port, Uživatelské jméno a heslo. Pro připojení lokálně na PC změním adresu serveru 78.80.123.80 za localhost. PostgreSQL je zobrazeno na Obr. 13 a MySQL na Obr. 14.



Obr. 12 – Nový zdroj dat v systému Windows

Obr. 13 – Nastavení ODBC ovladače pro připojení na Server PostgreSQL



Obr. 14 – Nastavení MySQL ovladače pro přístup na localhost

5.2 Tvorba skriptů

Skripty jsou vytvořeny v jazyce WinwrapBasic, který je založený na Visual Basic pro platformy .NET. Skripty mají koncovku BAS (Basic source code file).

Skripty komunikují s uživatelem pomocí Uživatelských dialogů. Při prvním spuštění skriptů máme dvě možnosti, uživatelský dialog nabídne načíst data z uloženého souboru s nastavením nebo vybrat ze kterých dat v databázi budeme tvořit graf. Při výběru druhé možnosti se skript dostane do cyklu, ve kterém vybírá data k načtení, barvu grafu, jména os a novou osu Y. Cyklus končí, pokud ho uživatel zruší nebo nedosáhne-li počet čar v grafu na 10. Po ukončení cyklu, je možné uložit nastavení vytvořeného grafu do souboru. Při první možnosti, skripty načítají z databáze podle nastavení v uloženém souboru. Dále jsou popsány důležité části skriptu.

5.2.1 Uživatelský dialog

Uživatelský dialog začíná příkazem `Begin Dialog` a končí `End Dialog`, vše obsažené uvnitř těchto příkazů se zobrazí v dialogu, musí se ale každému z příkazů nastavit pozice a velikost. Dialog použitý v práci je na Obr. 15.

```
Begin Dialog UserDialog 400,203 ' %GRID:10,7,1,1
    CancelButton 190,161,90,21
    Text 20,14,250,14,"Přejete si načíst uložené nastavení?",.Text1
    PushButton 50,56,90,21,"Ano",.PushButton1
    PushButton 200,56,90,21,"Ne",.PushButton2
    Text 50,98,260,14,"Stisknutím NE pokračujte výběrem grafu",.Text2
End Dialog
Dim dlg As UserDialog
Select Case Dialog (dlg)
```

Obr. 15 – Dialog uvedený ve skriptu pro načtení nastavení

Dialog se vyvolá pomocí `Dialog (dlg)`. Pro ošetření výstupů (tlačítko Cancel, křížek, OK, nebo jakákoliv další) používám Select Case konstrukci: `Select Case Dialog (dlg)`. V tomto případě může vracet 3 hodnoty: 0,1 a 2. Případ 0 když je zvoleno tlačítko Cancel nebo křížek, 1 kdy je zvoleno tlačítko Ano a 2 kdy je zvoleno tlačítko Ne. Ošetřovat jde i výstup z TextBoxů pomocí metody `dlg.Text1`.

5.2.2 Načítání dat z databáze

Pro načítání dat jsem používal metodu worksheetu `.LoadDB()`, jejíž parametry jsou: pozice, odkud ve worksheetu má začít načítat, připojovací řetězec a dotaz do databáze.

```
Set Worksheet1 = Grapher.Documents.Add(grfWksDoc)
pripojeni_DB = "DSN=VZD_server_david;DATABASE=postgres;SERVER=78.80.123.80;
dotazdb = "SELECT time,value FROM data WHERE sensor_number =" + Str(dodb)
Worksheet1.LoadDB(1,1,pripojeni_DB,dotazdb)
```

Do řetězce `pripojeni_DB` přiřazujeme připojovací řetězec pro PostgreSQL databázi s názvem VZD_server_david. Další použitý řetězec obsahuje SQL dotaz do databáze. Načítáme čas a hodnotu za použití senzoru a bodu měření.

```
dodb = Left(dlg1.combo$, 1) 'senzor z DB
dodb2 = Right(dlg1.combo$, Len(dlg1.combo$)-2) 'Bod měření z DB
```

Obr. 16 – senzor a bod měření použitý v SQL dotazu

V klauzuli WHERE používám číslo senzoru a bod měření z komboboxu v uživatelském dialogu, `dodb` a `dodb2`. Viz Obr. 16

5.2.3 Načtení nastavení

Nastavení je textový soubor, kam se ukládají hodnoty při práci s částí skriptu, která se připojuje do databáze a stahuje data. Načtení provádím procházením souboru a rozložením řádků do pole, ze kterého sestavuji graf s aktuálními hodnotami z databáze, který byl uložený v nastavení.

```
On Error GoTo Problem
Dim nastaveni(100) As String
pocet_nastaveni = 0
soubor = GetFilePath$()
Open soubor For Input As #1
While Not EOF(1)
    Line Input #1, L
    Debug.Print L
    nastaveni(pocet_nastaveni) = L
    pocet_nastaveni = pocet_nastaveni + 1
Wend
Close #1
```

Při chybě čtení souboru nebo při vybrání jiného souboru, je nastavena výjimka, která chybu popíše a znovu vrátí k výběru souboru. Pomocí `Soubor = GetFilePath()` vybírám soubor pro čtení. `Debug.Print L` slouží k výpisu do konzole, které řádky se ukládají do pole. Na konci čtení soubor uzavírám `Close #1`.

5.2.4 Vytváření grafů

V načítání nastavení vytvářím graf z hodnot, která jsou uložené v poli (Jednoduché vytvoření grafu je v kapitole 2.4.1). Vytvořím si nový worksheet a do něj načítám data z databáze. V `nastaveni(0)` je uložen připojovací řetězec a v `nastaveni(1)` dotaz do databáze.

```
Worksheet1.LoadDB(1,1,nastaveni(0),nastaveni(1))
Jméno = Worksheet1.Name()
delka = Len(Jméno)
Debug.Print Jméno
Set Graf_tunel = Grapher.Documents.Add(grfPlotDoc)
Graf_tunel.Shapes.AddLinePlotGraph(Left(Jméno,delka-2),1,2)
Set Graph1 = Graf_tunel.Shapes.Item(1)
Set YAxis1 = Graph1.Axes.Item(2)
Set LineScatterPlot1 = Graph1.Plots.Item(a)
Set XAxis1 = Graph1.Axes.Item(1)
Set YAxis1 = Graph1.Axes.Item(2)
XAxis1.title.text = nastaveni(2)
YAxis1.title.text = nastaveni(3)
LineScatterPlot1.line.foreColor =  nastaveni(4)
```

Obr. 17 – vytvoření grafu z hodnot uložených v poli při načítání nastavení

`Jméno` a `delka` jsou proměnné, které používám pro skutečné jméno worksheetu v programu Grapher, aniž bych ho musel někam ukládat a poté načítat. Kdykoliv se ve worksheetu něco změní, k jeho názvu se přidá * a vypadá takto: „worksheet *“. Příkaz `Left(Jméno,delka-2)` tvoří název bez mezery a * a mohu tak pracovat s otevřeným neuloženým souborem. Nastaví se Graph1, do něj se přidají 2 osy (Je-li

v uloženém nastavení i další osa Y, přidá se do grafu další Y osa pro čáru), čára, popis os a barva čáry. V dalších krocích se přidávají do grafu čáry s barvou a legenda.

5.3 Použití skriptů

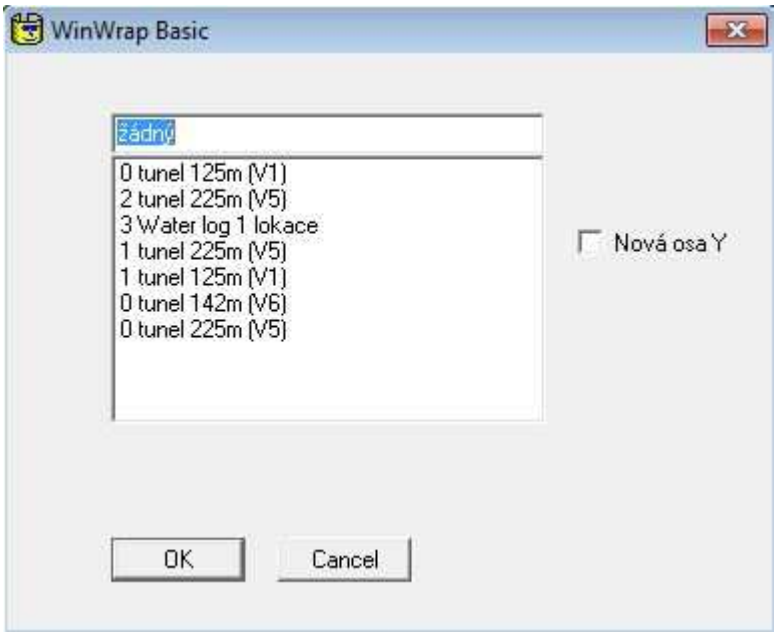
V této části se je vysvětleno použití vypracovaných skriptů. Skripty jsou nastaveny na použití databáze PostgreSQL. Pro použití databází Microsoft Access a MySQL je nutné změnit řádky které obsahují část připojení k databázím, ve skriptu wksheet.BAS,

Pro práci se skripty musíme načíst skript UD_start.BAS ve skript Manageru programu Grapher. Spustíme ho tlačítkem start/resume ve Script Manageru.

Obr. 18 – Načtení uloženého nastavení. První zobrazený dialog Scriptu.

V prvním Uživatelském dialogu máme na výběr 2 možnosti (pokud nepočítáme zrušení spuštěného skriptu tlačítkem Cancel nebo křížkem), první možností je tlačítko Ano, která spouští skript Nacteni_nastaveni.BAS a vytvoří graf z uloženého nastavení. Po stisknutí tlačítka budeme dotázáni pro výběr souboru s uloženým nastavením. Uložené nastavení je vždy textový soubor, uživatel si ho může pojmenovat dle vlastního uvážení. Skript pokračuje vykreslením grafu, pokud jsme použili na nahrání jiný typ souboru, skript zobrazí chybovou hlášku a vyzve nás ke znovu načtení souboru. Po načtení grafu je skript ukončen.

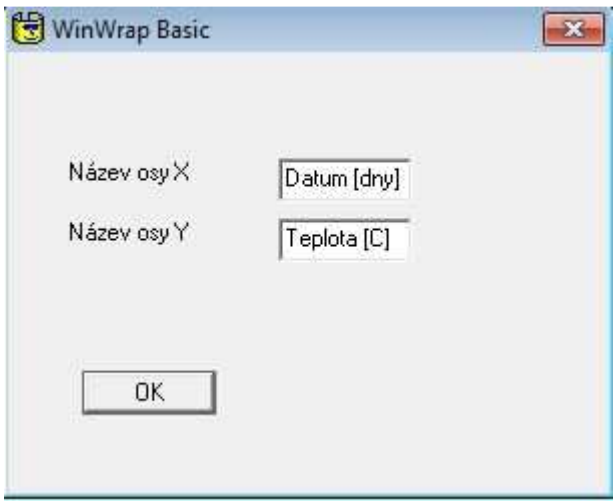
Tlačítko Ne spustí skript wksheet.BAS. Zobrazí se UD pro výběr senzoru z databáze, spustí se cyklus ve kterém přidáváme čáry do grafu.



Obr. 19 - Výběr senzoru s bodem měření, můžeme vybrat i jestli v grafu vytvoříme novou osu Y

V dialogu pokračujeme výběrem senzoru, máme možnost si vybrat z 8 senzorů. Výběrem „žádný“, ukončíme výběr senzorů, stejně jako tlačítkem Cancel. Máme zde možnost zatrhnout výběr Nová osa Y, tím vytvoříme novou osu Y a bude pro nás jednodušší porovnávat grafy s rozdílnými hodnotami. Přidaných čar do grafu může být až 10.

Po vybrání prvního senzoru ze seznamu se nám zobrazí dialog, ve kterém zadáme názvy os. Tento Dialog se zobrazuje pouze jednou, po vybrání prvního senzoru.

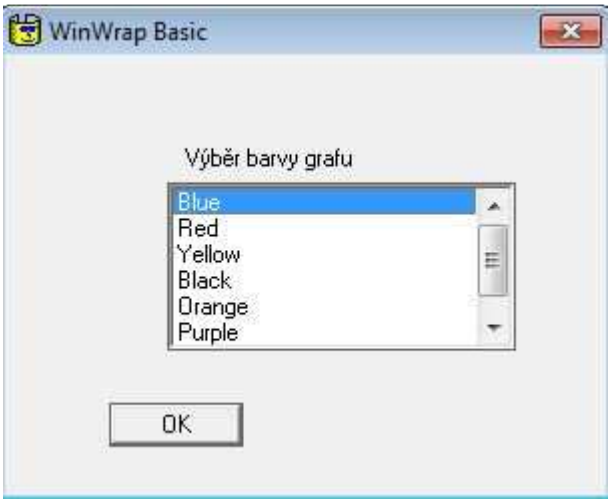


Obr. 20 – Dialog, kde se zadávají názvy os X a Y

Zadané názvy skript uloží a přiřadí je k osám X a Y v grafu.

Po vyplnění názvů os se dostaneme k výběru barvy čáry v grafu. Můžeme vybrat 8 různých barev. Po vybrání barvy skript obarví čáru v grafu a znovu se zobrazí

Uživatelský dialog s výběrem senzoru a bodu měření. Vracíme se na začátek cyklu k výběru senzoru a bodu měření.



Obr. 21 – Dialog s výběrem barvy grafu. Je možné vybrat 8 různých barev

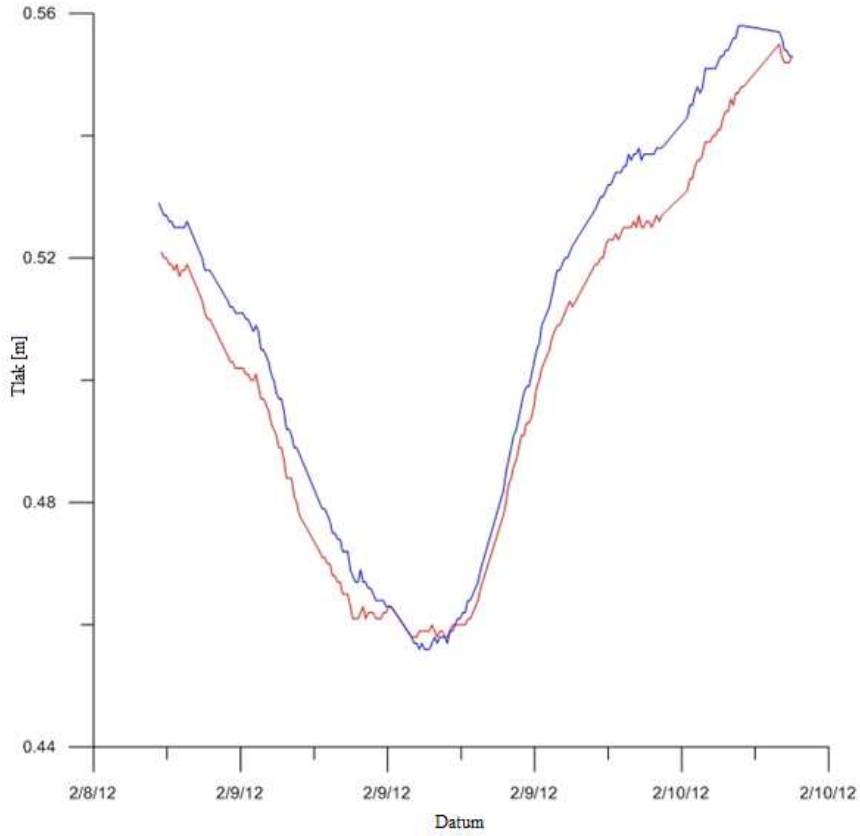
Poté co si ve výběru senzorů vybereme žádný a budeme chtít skript ukončit, zobrazí se předposlední dialog, který se nás zeptá, zda chceme uložit nastavení vytvořeného grafu (samotný graf nemá smysl ukládat, když se data v databázi mění, proto ukládáme nastavení). Po tomto kroku je do grafu přidána i legenda.

Obr. 22 – Dialog, s názvem uloženého nastavení do textového souboru

Název souboru je ošetřený proti zapsání nesmyslného souboru, kdy skript pomocí chybové hlášky oznámí, jakou chybu jsme při zadávání souboru udělali a znovu si vyžádá zadání jména souboru.

5.3.1 Ukázky grafů

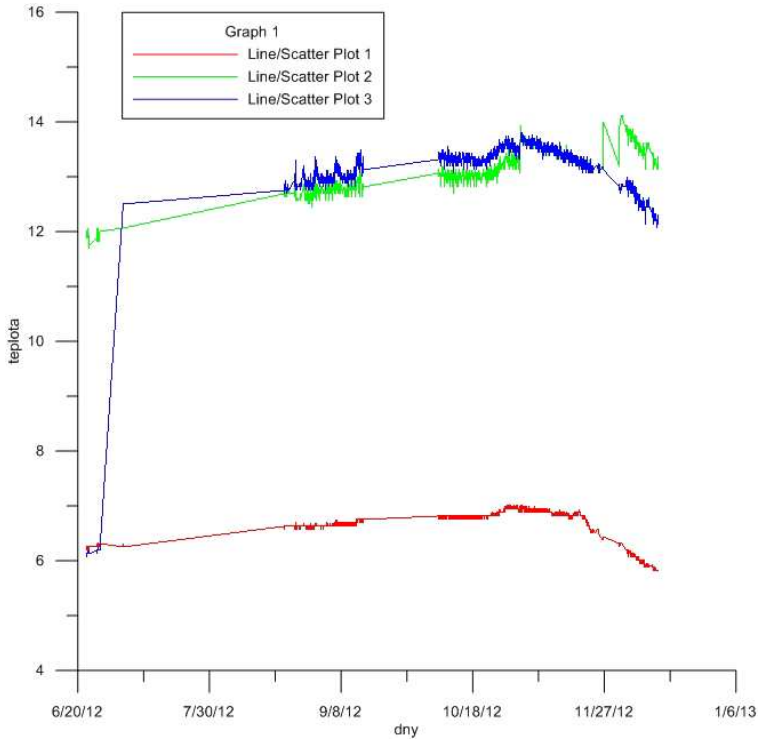
V této části jsou zobrazeny ukázky grafů z hodnot stažených z databáze Tunelu Bedřichov, více o obsahu měření v kapitole 4.2. V této části se také zabývám rychlostí načítání dat z databáze pro různé objemy dat do programu Grapher.



Obr. 23 - Graf porovnání dvou čar v grafu s objemem dat pro čáru 700 záznamů

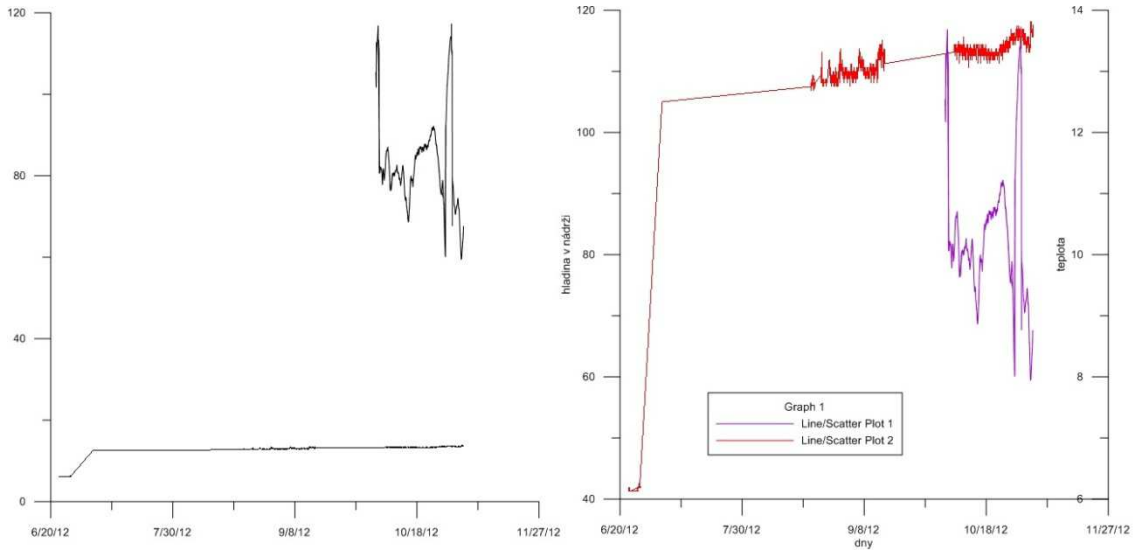
Na Obr. 23 vidíme výsledný graf po načtení ze dvou senzorů. V databázi bylo přes 700 záznamů k jednotlivým senzorům. Graf se zobrazuje s hodnotami od 0,45 až po 0,55 v době od 8.2.2012 až do 10.2.2012, kdy se záznamy do databáze ukládaly po deseti minutách. Doba načtení databáze byla přibližně 7 sekund. Data byla načtena z databáze Microsoft Access za použití naprogramovaných skriptů.

Obr. 24 zobrazuje tři čáry v grafu vytvořeného pomocí naprogramovaných skriptů. Ke každé z čar se v databázi vztahuje 23300 záznamů a do databáze byly odesílány ve stejném časovém rozmezí. Čas načtení databáze se pohyboval přibližně minutu. Data byla načtena z databáze PostgreSQL.



Obr. 24 – Vložení tří čar do grafu s obsahem dat v databázi pro čáru přes 23000 záznamů

Na dalším obrázku Obr. 25 je porovnání grafů s/bez použití naprogramovaných skriptů. Graf vlevo je vytvořen ručně a vpravo je vytvořen pomocí naprogramovaných skriptů: graf s dvěma osami, popiskami os, legendou a zabarvenými čárami. Pro každou čáru bylo v databázi přes 11000 záznamů a doba načítání se pohybovala přibližně 30 sekund. Data byla načtena z databáze PostgreSQL



Obr. 25 – porovnání vytvořených grafů s/bez použití naprogramovaných skriptů

Z hlediska rychlosti pro načítání dat z databáze do programu Grapher jsou nejrychleji načtena data s malým obsahem dat. Objem dat do 1000 záznamů je načten do 10

sekund, data s objemem dat do 10000 záznamů jsou načtena do 30 sekund. Data s objemem do 20000 záznamů jsou načtena do minuty. Při načítání velkého objemu dat, nad 10000 záznamů se stane, že program Grapher není schopen reagovat na nic jiného, než na načítání dat, takže nelze probíhající načítání zrušit ani nijak jinak s programem pracovat. Po načtení dat do programu s ním lze opět pracovat.

6 Závěr

V bakalářské práci jsem navrhl skripty, které automatizují práci s programem Grapher. Skripty se pomocí databázového rozhraní připojují k databázi na serveru a stahují si z ní data, ze kterých automatizují tvorbu grafu. Skripty dokážou načíst obsah databáze a porovnávat data z více senzorů v jednom grafu. Při porovnávání dat z více senzorů, jsou pak čáry barevně odlišeny a počet zobrazených čar je maximálně 10. Skripty mohou ukládat a načítat nastavení vytvořeného grafu, což umožňuje získání stejného grafu z aktuálních hodnot v databázi.

Během práce jsem setkal s chybou v automatickém připojení do databáze, k databázi PostgreSQL v programu Grapher, kterou se mi při řešení v databázi MySQL podařilo nahradit a poté jsem úspěšně dokončil připojení do PostgreSQL databáze a nyní již vytvořené skripty úspěšně stahují data z databáze PostgreSQL na serveru.

Použití skriptů je možné pouze na 32 bitových operačních systémech. 64bitové verze systému nepodporují vytvořené databázové rozhraní.

Program se může i dále upravovat a vyvíjet podle potřeb vedoucího bakalářské práce při jeho práci s programem Grapher a daty z databáze tunelu Bedřichov.

Seznam použité literatury

[1] *AccuTerm 2K2: Scripting Language Reference Manual.* Sunland, CA, USA: AccuSoft Enterprises, 2003.

[2] Bedřichovský tunel. [online]. Liberec: Technická univerzita v Liberci. [cit. 15. 11. 2011] URL: <http://bedrichov.tul.cz/Tunel/index.php>

[3] BĚLOCH, M. *Programy Grapher, MS Excel a MatLab a možnosti jejich připojení k databázi prostřednictvím rozhraní ODBC.* [projekt] Liberec, Technická univerzita v Liberci, 2012.

[4] *Databáze.* [online]. Wikipedia.org [cit. 29. 3. 2012] URL: <http://cs.wikipedia.org/wiki/databáze>

[5] *Grapher User's Guide, 2D & 3D Graphing Software for Scientists, Engineers & Business Professionals*. Golden, CO, USA: Golden Software, Inc., 2011.

[6] HOKR, M. a kol. *Tunel Bedřichov: Charakterizace granitoidů in situ.* [závěrečná zpráva]. Praha: SÚRAO, 2010.

[7] HOKR, M. a kol. *Zpráva pro 2. kontrolní den projektu TUNEL 2011*. Liberec: Technická univerzita v Liberci, 2011.

[8] MUKNŠNÁBL, J. *ODBC v kostce.* [online]. Reboot.cz [cit. 30. 4. 2012] URL: <http://reboot.cz/howto/databaze/odbc-v-kostce/articles.html?id=152>

[9] *Relační databáze.* [online]. Wikipedia.org [cit. 29. 3. 2012] URL: <http://cs.wikipedia.org/wiki/Relační_databáze>

[10] SVOBODA, P. *Zařízení pro vzdálený sběr a přenos dat – Firmware.* Liberec: Technická univerzita v Liberci, 2011.

[11] *The PostgreSQL Global Development Group* [online]. PostgreSQL.org [cit. 25.11.2012] URL:< http://www.postgresql.org/docs/9.2/static/tutorial.html>

Seznam příloh

Příloha A – Seznam použitých měřících zařízení v Tunelu Bedřichov

Příloha B – Vytvořené skripty a nakonfigurované ODBC ovladače pro PostgreSQL a pro Microsoft Access přiložena vytvořená databáze (umístěny na CD).

Příloha A – Seznam použitých měřících zařízení v Tunelu Bedřichov

Pramen	Automatické měření, frekvence záznamu				Popis
	výtok	Teplota	pH	redox	
AV1	sklopka Fiedler 5 min	…	…	…	žlábek v metráži 125 m od portálu
AV2	Lapač srážek se sklopkou MicroLog T3 každé sklopení	Teplotní čidla Comet 10 min	…	…	dva žlábky v metráži 798,5 m (vzdálenější má č. V 2/1 blíže k portálu a o něco níže ve stěně je V 2/2). Žlábky jsou na dvou různých drobných puklinách ve vzdálenosti 30 cm
AV3	Sklopka Comet 10 min	…	…	…	žlábek umístěný na metráži 1375,5 m
AV4-1		Teplotní čidla Comet 15 min	…	…	dvě hadice v místech zabetonovaných puklin ve stropě v pravém boku tunelu A v metráži 1728,5 m. V4/1 je umístěna výše ve stropě a V4/2 o něco níže
AV4-2	Hladinoměr Levelogger 5 min	…	…	…	
AV5	sklopka Modul infrastr. 5 min	…	Sonda theta90 Modul in 5 min	…	ze stěny tryskající pramen, zjištěný je v roce 2006 v metráži 226,4 m. V roce 2007 poklesu jeho výtoku byl opatřen žlábkem
AV6	Sklopka Modul infrastr. 5 min	…	…	…	výtok ze starší železné roury v pravém boku v metráži 142 m
AV7	hladinoměr Levelogger 15 min	…	…	…	hadice v metráži 76,5 m, je to čtvrtá hadice od portálu z pěti, která má vždy nejsilnější výtok z okolních hadic
AV8	…	…	…	…	celkový výtok z potrubí sbírající vodu z celého tunelu na u portálu tunelu A
AV10	…	…	…	…	výtok ze sběrného potrubí na metráži 2420m
W67	…	…	…	…	hadice v místě zabetonované pukliny v pravém boku tunelu na metráži 67 m
W142					hadička ústící z navrtané pukliny v pravém boku tunelu na metráži 142 m (sousedí s pramenem V6)
W1565					přehrazení skalní prohlubně v levém boku tunelu na metráži 1565 m, vývod realizován hadičkou; pro

					odběr vzorků bez přístupu vzduchu slouží další hadička zapuštěná dovnitř skály
W2210					do žlabu svedený výtok z upraveného otvoru v betonovém nástřiku, levý dolní bok tunelu na metráži 2210 m
W2313					hadice v místě zabetonované pukliny na metráži 2313 m v pravém boku tunelu
W2470	Hladinoměr				výtok ze staré železné roury v levém boku tunelu na metráži 2470 m
	Levelogger 15 min				
Kanál 2470m	…	Teplotní čidla	…	…	
		Comet 15 min	…	…	
Kanál 100m	Výška hladiny, ultrazvukové čidlo US1200	Teplotní čidlo	Čidlo pH	Čidlo ORP	